

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування та спеціалізованих комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ Віталій РОМАНКЕВИЧ

«__» _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Системне програмування та спеціалізовані
комп'ютерні системи»**

спеціальності 123 Комп'ютерна інженерія

на тему: «Комп'ютерна система тестування з розпізнаванням підпису»

Виконав:

Студент IV курсу, групи KB-61

Шуть Максим Вікторович _____

Керівник:

Доцент кафедри СПСКС, к.т.н., с.н.с.,

Боярінова Юлія Євгенівна _____

Консультант з нормоконтролю:

Доцент кафедри СПСКС, к.т.н.,

Клятченко Ярослав Михайлович _____

Рецензент:

Зав. відділом _____, к.т.н, с.н.с.,

Андрієнко Андрій Андрійович _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра системного програмування та спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ Віталій РОМАНКЕВИЧ

«___» _____ 2020 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Шутю Максиму Вікторовичу

1. Тема проєкту «Комп'ютерна система тестування з розпізнаванням підпису»,
керівник проєкту Боярінова Юлія Євгенівна, к.т.н., доцент, с.н.с.,
затверджені наказом по університету від «25» травня 2020 р. №1181-С.
2. Термін подання студентом проєкту: «___» _____ 2020 р
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - огляд існуючих підходів, щодо комп'ютерної системи тестування з розпізнаванням підпису;
 - проектування комп'ютерної системи тестування з розпізнаванням підпису;
 - реалізація комп'ютерної системи тестування з розпізнаванням підпису;

5. Перелік графічного матеріалу:

- узагальнена структура видозміненого персептрона (схема структурна);
- розпізнавання підпису нейронною мережею (схема алгоритму);
- архітектура комп'ютерної системи тестування з розпізнаванням підпису (схема структурна);
- функціонування нейронної мережі для розпізнавання підпису (схема алгоритму).

6. Консультанти:

Питання	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к.т.н., доцент каф. СПСКС		

7. Дата видачі завдання: “23” жовтня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою проєкту	07.11.2019	
2.	Розроблення та узгодження технічного завдання	15.11.2019	
3.	Аналіз існуючих рішень	31.12.2019	
4.	Підготовка матеріалів першого розділу проєкту	13.01.2020	
5.	Розроблення програмного забезпечення	17.02.2020	
6.	Відлагодження програмного продукту	02.03.2020	
7.	Підготовка матеріалів другого розділу проєкту	31.03.2020	
8.	Розроблення інтерфейсу програмного забезпечення	11.04.2020	
9.	Підготовка графічної частини	19.05.2020	
10.	Оформлення документації дипломного проєкту	20.05.2020	
11	Попередній розгляд дипломного проєкту на кафедрі	20.05.2020	

Студент

Максим ШУТЬ

Керівник проєкту

Юлія БОЯРІНОВА

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (54 с., 20 рис., 7 табл., 6 додатків, 8 бібліогр.).

Сьогодні процес автоматизації контролю знань є необхідним. Тому актуальність даного дипломного проєкту полягає саме в розробці комп'ютерної системи тестування з розпізнаванням підпису.

Метою дипломного проєкту є покращення сучасного стану тестового контролю знань.

Об'єктом дослідження є процес отримання графічних відповідей у системі комп'ютерного тестування з розпізнаванням підпису.

Предметом дослідження є програмний засіб комп'ютерного тестування з розпізнаванням підпису.

Після виконаних теоретичних та практичних досліджень розроблено комп'ютерну систему тестування з розпізнаванням підпису. Вона виконує функцію перевірки засвоєння та практичного застосування знань, одержаних студентами (учнями) під час навчального процесу. Також програмна розробка надає можливість ідентифікувати особистість людини, яка проходила щойно тестування, шляхом розпізнавання її рукописного підпису за допомогою багатошарової нейронної мережі. В результаті роботи програми користувач отримує кількість правильних відповідей на запитання і власне прізвище в друкованому вигляді. Слід зазначити, що розроблена програма в сукупності з доволі потужним функціоналом, має досить зручний та зрозумілий інтерфейс.

Ключові слова: комп'ютерна система тестування, багатошарова нейронна мережа, розпізнавання підпису, програмна розробка, ідентифікація особистості, рукописний підпис, практичне застосування знань, автоматизація, перевірка засвоєння знань, потужний функціонал, інтерфейс.

ANNOTATION

Diploma project includes an explanatory message (54 p., 20 pic., 7 tabl., 6 applications, 8 bibliogr.).

Today, the process of automating knowledge control is necessary. Therefore, the relevance of this diploma project lies in the development of a computer-based testing system with signature recognition.

The aim of the diploma project is to improve the current state of test control of knowledge.

The object of research is the process of obtaining graphical answers in a computer-based test system with signature recognition.

The subject of the study is a software tool for computer testing with signature recognition.

After theoretical and practical research, a computer-based testing system with signature recognition was developed. It performs the function of checking the assimilation and practical application of knowledge acquired by students (pupils) during the educational process. Software development also makes it possible to identify a person who has just been tested by recognizing his or her handwritten signature using a multilayer neural network. As a result of the program, the user receives the number of correct answers to questions and their own name in print. It should be noted that the developed program in combination with a fairly powerful functionality, has a very user-friendly and clear interface.

Keywords: computer testing system, multilayer neural network, signature recognition, software development, identity, handwritten signature, practical application of knowledge, automation, knowledge acquisition testing, powerful functionality, interface.

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до програмного продукту, що розробляється.....	3
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного та апаратного забезпечення користувача.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.045490.002 ТЗ							
Змін.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система тестування з розпізнаванням підпису. Технічне завдання					Літ.	Аркуш	Аркушів
Розробив	Шуть М.В.											
Перевірив	Боярінова Ю.Є.										1	4
										КПІ ім. Ігоря Сікорського, ФПМ, КВ-61		
Н. контроль	Клятченко Я.М.											
Затвердив	Романкевич В.О.											

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Комп'ютерна система тестування з розпізнаванням підпису».

Галузь застосування: проведення тестувань за різноманітними темами з ідентифікацією особистості, яка проходить тестування, шляхом розпізнавання її рукописного підпису.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення програмного забезпечення для перевірки зросту знань студентів (учнів) викладачами (вчителями) у вигляді тестувань з розпізнаванням підпису, а також покращення в цілому сучасного стану тестового контролю знань

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

					ІАЛЦ.045490.002 ТЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- наявність простого, зрозумілого та зручного інтерфейсу для користувача будь – якої вікової категорії;
- наявність випадкової генерації тестових питань;
- відповіді на запитання повинні бути представлені у вигляді рукописних символів за допомогою комп'ютерної миші ;
- розроблена програма повинна розпізнати введену відповідь та порівняти її з правильною за допомогою бази даних відповідей;
- під час проходження тестування повинен здійснюватися підрахунок правильних відповідей;
- можливість введення свого підпису за допомогою комп'ютерної миші ;
- розроблена програма повинна розпізнати підпис, тим самим ідентифікувати особистість, яка щойно проходила тестування;
- після завершення тестування користувач повинен отримати кількість правильних відповідей та власне прізвище у друкованому вигляді.

5.2. Вимоги до апаратного забезпечення

- оперативна пам'ять: 4 Гб;
- твердотілий накопичувач: 128 Гб.

5.3. Вимоги до програмного та апаратного забезпечення користувача

- операційна система: Windows 7/8/10;
- інтегроване середовище розробки ПЗ: Visual Studio.

					ІАЛЦ.045490.002 ТЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Видача завдання на дипломне проектування	29.10.2019
2.	Вивчення літератури за тематикою роботи	07.11.2019
3.	Розроблення та узгодження технічного завдання	15.11.2019
4.	Розроблення структури додатку	30.12.2019
5.	Розроблення дизайну та графічних елементів	14.01.2020
6.	Програмна реалізація додатку	16.02.2020
7.	Тестування додатку	03.03.2020
8.	Підготовка матеріалів текстової частини проекту	01.05.2020
9.	Підготовка матеріалів графічної частини проекту	19.05.2020
10.	Оформлення технічної документації проекту	20.05.2020
11	Попередній розгляд дипломного проекту на кафедрі	20.05.2020

[illegible]

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ _____	3
ВСТУП _____	4
1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ, ЩОДО КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПISУ _____	6
1.1 Огляд комп'ютерних систем тестування _____	6
1.2 Теоретичні основи розпізнавання зображень _____	11
1.3 Огляд існуючих методів для розпізнавання підпису _____	16
1.4 Розпізнавання тексту за допомогою нейронних мереж _____	20
Висновки до розділу _____	22
2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПISУ _____	23
2.1 Опис комп. системи тестування з розпізнаванням підпису _____	23
2.2 Проектування комп. системи з розпізнаванням підпису _____	32
Висновки до розділу _____	37
3 РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПISУ _____	38
3.1 Вибір типу архітектури ПЗ та методу проектування _____	38

					ІАЛЦ.045490.004 ПЗ							
Змін.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система тестування з розпізнаванням підпису. Пояснювальна записка					Літ.	Аркуш	Аркушів
Розробив		Шуть М.В.										
Перевірив		Боярінова Ю.Є.									1	54
Н. контроль		Клятченко Я.М.								КПІ ім. Ігоря Сікорського, ФПМ, КВ-61		
Затвердив		Романкевич В.О.										

3.2 Розробка архітектури комп'ютерної системи тестування з розпізнаванням підпису _____	41
3.2.1 Діаграма варіантів використання для комп.системи _____	42
3.2.2 Діаграма класів комп'ютерної системи _____	44
3.2.3 Діаграма станів комп'ютерної системи _____	46
3.3 Інтерфейс комп'ютерної системи тестування з розпізнаванням підпису _____	47
Висновки до розділу _____	52
ВИСНОВКИ _____	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ _____	54
ДОДАТКИ _____	55
КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ	
ДОДАТОК А _____	ІАЛЦ.045490.005 Д1
ДОДАТОК Б _____	ІАЛЦ.045490.006 Д2
ДОДАТОК В _____	ІАЛЦ.045490.007 Д3
ДОДАТОК Г _____	ІАЛЦ.045490.008 Д4
ДОДАТОК Д _____	Лістинг розробленої програми
ДОДАТОК Е _____	Презентація бакалаврського проєкту

ПЕРЕЛІК СКОРОЧЕНЬ

КСТЗ – Комп'ютерна система тестування знань

ПЗ – Програмне забезпечення

ПК – Персональний комп'ютер

СТЗ – Системи технічного зору

HTML – HyperText Markup Language(мова розмітки гіпертексту)

MOODLE – Modular Object-Oriented Dynamic Learning Environment

(відкрита система управління навчанням)

PDF – Portable Document Format(відкритий формат файлу)

OCR – Optical Character Recognition(оптичне розпізнавання символів)

OpenCV – Open Source Computer Vision Library(бібліотека функцій та алгоритмів обробки зображень)

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Проблема підвищення ефективності, зокрема, надійності контролю якості освіти в різноманітних навчальних закладах завжди була актуальною, що призводило до детального та серйозного ставлення до неї. В наш час проводиться як науково-дослідна, так і практична робота над розробкою та тестуванням різних комп'ютерних систем для тестування знань, що базується в цілому на нормативному тестовому контролі знань та вмінь усіх учнів [1].

Зауважимо, що сьогодні важливо автоматизувати процес контролю знань. Для досягнення цієї мети в даний час, зазвичай, використовують різні інноваційні технології, зокрема, комп'ютерні системи тестування з розпізнаванням підписів. За допомогою такого програмного забезпечення можна здійснювати регулярні перевірки засвоєння, а також практичного застосування знань повною мірою, здобутих студентами (учнями) під час навчального процесу [1].

Актуальність роботи полягає в розробці комп'ютерної системи тестування з розпізнаванням підпису. Отримання відповідей на тестові питання в даній комп'ютерній системі має здійснюватися на основі нарисованих комп'ютерною мишею відповідей, в результаті ми отримаємо кількість правильних нарисованих відповідей.

Метою дипломного проекту є покращення сьогочасного стану тестового контролю ерудиції, а також дослідження існуючих комп'ютерних систем перевірки знань та реалізація комп'ютерної системи тестування з розпізнаванням підпису [1].

Поставлена мета досягається розв'язанням таких основних задач:

- необхідно проаналізувати існуючі підходи систем комп'ютерного тестування;
- спроектувати комп'ютерну систему тестування з розпізнаванням підпису;

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		4

– розробити програмний засіб та показати як він працює на практиці.

Об’єктом дослідження є процес отримання графічних відповідей у системі комп’ютерного тестування з розпізнаванням підпису.

Предметом дослідження є програмний засіб комп’ютерного тестування з розпізнаванням підпису.

Інформаційна технологія дозволяє розпізнавати графічні символи як відповіді на тестові запитання, забезпечуючи ефективну обробку даних та швидкодію.

Практична цінність отриманих результатів полягає у підвищенні достовірності та ефективності процесу тестування в комп’ютерних системах тестування.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		5

1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ, ЩОДО КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПИСУ

1.1 Огляд комп'ютерних систем тестування

Зі швидким зростанням інформаційних технологій автоматизація процесу тестування за допомогою використання комп'ютерних систем для тестування знань стає неймовірно актуальною.

КСТЗ - це інформаційна система автоматичного тестування, що представлена у вигляді діалогу, участь в якому приймають людина, яка, відповідно, намагається скласти довільний тест, та сам комп'ютер. В обов'язки другого входить автоматичне обчислення результатів іспиту цього ж користувача, а також отримання підсумкових показників за довільними критеріями відносно інших людей, які раніше вже складали відповідний іспит [1].

Тестові питання можна формувати за допомогою різних комп'ютерних інструментів, включаючи довільні редактори та програмне забезпечення, що використовується для створення презентацій, та за допомогою використання різноманітних мов програмування, а також широкого спектру можливостей, що може запропонувати Інтернет [1].

З-поміж усього розмаїття програм, що використовуються для створення довільних тестувань, користувачу неймовірно важко швидко зорієнтуватися та вибрати саме ту, що, дійсно, задовольняла б усі його вимоги. Виходячи із цієї проблеми, було вирішено провести аналіз найбільш відомих програм, та як результат – для наглядності створити таблиці порівняльних характеристик за різними критеріями. Досліджуючи таке програмне забезпечення, як: “Znannya”, “Indigo”, “EasyQuizzy”, “Miy test”, “eTest”, “HotPotatoes”, “MultiTesterSystem”, “AnsTester”, “Moodle”, “ADSoftTester”, - було виділено лише три програми, що володіють доволі функціональним, зручним та зрозумілим інтерфейсом

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

одночасно, мають тільки позитивні відгуки від користувачів, є досить простими у використанні, а також відповідають більшості поставлених вимог. Отже, далі більш детально буде розглядатися саме таке програмне забезпечення, як: “Miy test”, “Znannya”, “Moodle” [1].

“Miy test” – цілковита система декількох програм, що дає можливість формувати та проводити різні комп'ютерні тестування, а також забезпечує збір та аналіз відповідей. Слід відмітити, що дане програмне забезпечення здатне визначити бал пройденого іспиту, спираючись на шкалу тесту.

Зазвичай, “Miy test” використовують викладачі (вчителі), щоб швидко організувати та провести тестування або ж іспити в будь-яких навчальних закладах. Зазначимо, що дане програмне забезпечення доволі зручне й просте у використанні [1].

“Miy test” підтримує наступні функції:

1. можливість обрання конкретного місця на площині рисунка;
2. можливість введення довільної послідовності чисел за допомогою клавіатури;
3. можливість вибору єдиної відповіді;
4. можливість переставляти букви в будь-якому порядку;
5. можливість введення довільної послідовності символів за допомогою клавіатури;
6. можливість вибору декількох відповідей одночасно;
7. можливість встановлювати відповідності в конкретних питаннях;
8. можливість встановлювати сам порядок складання іспиту;
9. можливість надавати відповідь на запитання у формі “так” або “ні”.

Налаштування тесту, запитання, зображення до запитань зберігаються в одному тестовому файлі. Дане програмне забезпечення не використовує бази даних, якісь недоречні файли. Для проходження одного тестування

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		7

використовується єдиний файл. Використання “Miy test” дозволяє проводити різноманітні види іспитів, починаючи локальними тестуваннями, та, закінчуючи мережевими [1].

За допомогою програмного забезпечення “Znappya” можна формувати тести за різними тематиками, а також, відповідно, здійснювати тестування, використовуючи персональні комп’ютери, в інформатичних аудиторіях шкіл, вищих навчальних закладів, або ж персонально вдома за допомогою ПК чи ноутбука.

Програмі “Znappya” притаманні наступні особливості [1]:

1. Оцінки після проходження тестування можуть бути представлені у вигляді двох основних форм: за допомогою цифр згідно зі шкалою від нуля до дванадцяти балів для школярів чи за допомогою добре відомих слів для студентів (“незалік” або ж “залік”);
2. Формування різних за складністю, оформленням та варіантами відповідей тестів;
3. Результат тестування у вигляді звіту можна дуже швидко та просто зберегти в довільному файлі або ж відразу роздрукувати;
4. Для формування тестів можна використовувати різноманітні рисунки, якісь допоміжні текстові дані, що підтримують функцію HTML-форматування, необхідні файли, довільні графіки чи таблиці.

Додатковою особливістю даного програмного забезпечення є наявність тестового сервера, за допомогою якого вчитель (викладач), який організовує іспит в інформатичному класі школи чи аудиторії університету із використанням одночасно декількох комп’ютерів, може легко контролювати хід проходження іспиту учнями (студентами) за допомогою свого комп’ютера [1]. Моніторинг здійснюється за кожним комп’ютером без виключень. Вчителю (викладачу)

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8

надається можливість відстежувати поточний відсоток проходження іспиту окремо кожного учня (студента) або ж всіх разом, число правильних та, відповідно, неправильних відповідей на запитання тесту, а також їх відсоткове співвідношення.

“Moodle” являє собою повністю безкоштовну систему, що, зазвичай, використовується саме для електронного навчання [1].

Дана програма встановлюється саме на віддаленому сервері. Таким чином, будь-які користувачі можуть з легкістю отримати доступ до запропонованого сервера за допомогою довільного браузера, використовуючи підключення до мережі Інтернет.

“Moodle” підтримує наступні функції [1]:

1. можливість розміщення запитань тесту в довільному порядку;
2. можливість проходження одного іспиту декілька разів за дозволом вчителя (викладача);
3. можливість генерації запитань тесту за допомогою бази даних випадковим чином;
4. можливість вибору єдиної відповіді;
5. можливість встановлення обмеження в часі для проходження іспиту;
6. можливість вибору декількох правильних відповідей одночасно;
7. можливість отримувати числові відповіді в конкретних питаннях;
8. можливість встановлювати відповідності в конкретних питаннях;
9. можливість надавати відповідь на запитання у формі “так” або “ні”;
10. можливість надавати підказки під час проходження іспиту;
11. можливість встановлювати послідовність в конкретних питаннях.

Порівняльний аналіз, обраних для дослідження раніше програм, за головними критеріями продемонстровано за допомогою таблиць 1.1 – 1.3.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		9

Таблиця 1.1 - Формування тесту

Критерії порівняння	“Miy test”	“Znannya”	“Moodle”
Імпорт тестів із текстових	+	+	+
Експорт тестів у текстові	+	-	+
Розбиття тестових завдань на	+	-	+
Перевірка правопису тексту	-	-	-
Форматування тексту	+	+	+

Таблиця 1.2 - Вставка об'єктів у тест

Критерії порівняння	“Miy test”	“Znannya”	“Moodle”
Графічний формат	+	+	-
Звуковий формат	-	-	-
Відеоформат	-	-	-
Підтримка анімації	-	-	-
Використання редактора формул	-	-	-
Вставка таблиць	+	+	-
Вставка діаграм	+	+	-

Таблиця 1.3 - Налаштування параметрів тесту

Критерії порівняння	“Miy test”	“Znannya”	“Moodle”
Обмеження часу тестування	+	+	+
Обмеження кількості тестувань	+	+	+
Обмеження часу відповіді на кожне тестове завдання	+	+	+
Перемішування тестових завдань	+	-	+
Випадковий вибір тестових завдань із тестової бази	+	-	+
Альтернативне оцінювання	+	+	+
Можливість задавання шкали оцінювання	+	+	+
Відкритість/закритість тестування	+	+	+

1.2 Теоретичні основи розпізнавання зображень

Обробка зображень для їх подальшого розпізнавання є одним з основних та важливих завдань у галузі розробки комп'ютерних систем тестування з розпізнаванням підпису.

Дана проблема має складний ієрархічний характер, що, в свою чергу, зумовлює виникнення ряду таких основних стадій, як: нормалізація обраних об'єктів, сегментація, розпізнавання, а також сприйняття області зору. Слід відмітити, що наймовірно важливий етап розуміння, іншими словами інтерпретація образів, частково включається в стадію сегментації, а остаточно вирішується вже на стадії розпізнавання [2].

Під час розробки СТЗ, а також безпосереднього порівняння зображення з еталонами, може з'явитися низка наступних специфічних проблем й труднощів:

1. стандарти, тут маються на увазі еталони, та образи можуть іноді відрізнятися геометричними перетвореннями;
2. відмінності між вхідними та еталонними зображеннями виникають через зміни освітлення, якихось локальних перешкод чи підсвічування;
3. вхідні образи не відповідають стандартам (еталонам) через випадкові перешкоди.
4. еталонні та вхідні зображення відрізняються розташуванням в області зору;
5. образи представлені на складному для сприйняття тлі.

Для вирішення даної проблеми в цілому, а також на окремих її стадіях використовуються доволі часто різноманітні процедури нормалізації, розпізнавання й сегментації [2].

Для наглядності класифікація головних методів обробки, а також розпізнавання за допомогою СТЗ зображень, продемонстрована на рисунку 1.1.



Рисунок 1.1 - Основні процедури і методи розпізнавання зображень

Операція попередньої обробки використовується майже завжди після вилучення інформації з відео датчика і спрямована на зменшення кількості шуму на картинці, що виникає в результаті вибірки та квантування й зменшення впливу якогось зовнішнього довільного шуму [2].

Під сегментацією мається на увазі процес пошуку однорідних ділянок на картинці. Даний крок є доволі складним і загалом не повністю алгоритмізованим для довільних образів. Найпоширеніші процедури сегментації сьогодні, зазвичай, базуються на визначенні однорідності яскравості певних кольорів чи однорідності цілковитих текстур.

За наявності стійких відмінностей у яскравості окремозятих ділянок області зору, зазвичай, використовуються порогові процедури. Метод забудови ділянок буде ефективним лише у випадку наявності стабільної сполученості в межах окремих сегментів. Процедуру виділення контуру слід використовувати у випадку, якщо вони є стійкими та чіткими одночасно. Ці методи використовуються для вибору сегментів за критерієм рівномірності яскравості. Слід зазначити, що один із найбільш ефективних методів забудови ділянок надає можливість вибору вхідних точок, як в автоматичному режимі, так і за допомогою оператора [2].

Взагалі усі методи є цілком прийнятними з точки зору обчислювальних витрат, однак для кожного з них характерна неоднозначність маркування точок у реальних ситуаціях через необхідність використання евристики, тут мається на увазі, вибір цифрових масок, порогів відповідності яскравості, тощо. У зв'язку з цим варто відзначити запропонований метод багатозначної розмітки, заснований на поєднанні різних методик для зменшення невизначеності. Звичайно, що провідне практичне значення мають такі процедури, що дозволяють алгоритмам паралельної обробки прискорити процес розмітки за допомогою логічного аналізу елементів, що розташовуються поруч [2].

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		13

Для опису та сегментації таких особливостей образів, як: регулярність, однорідність, шорсткість, - використовуються текстурні процедури, які, в свою чергу, можна розділити на дві великі групи – структурні та статистичні. Прикладом других є застосування матриць збігів, утворених зі вхідних образів, з подальшим обчисленням статистичних моментів. Наприклад, за допомогою перших на основі мозаїки Вороного можна побудувати багато полігонів. В одному регіоні знаходяться, об'єднані багатокутники, із загальними властивостями. Для вивчення загальних особливостей, зазвичай, використовують ознаки, тут маються на увазі моменти багатокутників [2].

Після сегментації можуть з'явитися шуми у вигляді розрізнених змін ізольованих елементів рисунка або ж у вигляді спотворень деяких сполучених ділянок. Не вдаючись докладно про методи боротьби з таким шумом, слід зазначити лише той факт, що на практиці найпоширенішими є цифрові фільтри - маски та нелінійні фільтри, такі як медіани. При сегментації шляхом вибору контурів, використання фільтрів усереднення - маски неможливо, так як вони не виділяються, а розмиваються. Для підкреслення контурів, інколи, використовуються спеціальні оператори цілісного типу, проте інтегральні є більш ефективними [2].

Розпізнавання часто є завершальною стадією обробки, яка є, безпосередньо, основою для процесів розуміння та інтерпретації. Вхідні дані для розпізнавання – це образи, відібрані в результаті стадії сегментації та частково відновлені. Вони можуть відрізнятися від еталонних, перш за все, збереженим шумом [2]. Інколи різними геометричними спотвореннями чи довільними спотвореннями яскравості.

На практиці для розпізнавання зображень, зазвичай, використовують чотири підходи, які, в свою чергу, базуються на різних методах. Слід зазначити, що кожен із них має право на існування. Перш за все, це зумовлено тим, що абсолютно кожний підхід характеризується своїми специфічними алгоритмами.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		14

Методи кореляції широко використовуються для виявлення, а також розпізнавання зображень у навігаційних системах, різноманітних промислових роботах, спеціалізованих системах стеження. З повністю заданим еталоном багатоступенева кореляція за допомогою сканування початкової області зору, по суті являє собою повноцінний пошук в площині сигналів. Таким чином, саме цей метод можна вважати базовим, найбільш стійким до різноманітних шумів, хоча водночас він є і найбільш трудомістким [2].

Інші процедури спрямовані на зменшення обчислювальних витрат, намагаючись забезпечити заздалегідь задану надійність розпізнавання, розмірних і вагових характеристик комп'ютера, витрат на програмне забезпечення, а також обладнання.

Ознакові та синтаксичні процедури, в свою чергу, вважаються найрозвиненішими в теорії розпізнавання образів. Вони базуються одночасно на детермінованому й статистичному підходах. Основна складність ознакових процедур - це вибір особливостей. Це засновано на природних правилах [2]:

- по-перше, сукупність ознак має бути найбільш мінімальною, бо від їх числа прямопропорційно залежить сама складність обробки в цілому, а також надійність;
- по-друге, особливості картинок одного класу можуть відрізнятися лише незначною мірою;
- по-третє, особливості картинок різних класів повинні відрізнятися повною мірою.

Хоча й процедури, засновані на використанні характеристик, не мають такої шумозахищеності, як процедури кореляції, через менші трудові витрати, їх корисно використовувати на першому етапі, коли вирішується проблема нееквівалентності. У цьому випадку деякі початкові образи, характеристики яких не відповідають жодному зі стандартів, відкидаються негайно [2].

Процедури нормалізації знаходяться поміж алгоритмами кореляції та алгоритмами, що засновані на використанні характеристик. На відміну від других при нормалізації образ ніяким чином не «втрачається», а лише замінюється образом того ж класу еквівалентності. З іншого боку, на відміну від перших багато початкових образів заміщаються багатьма нормалізованими образами. Кожен нормалізований образ, в загальному, значною мірою наближений до свого еталону (з точки зору групових перетворень), що в результаті неймовірно зменшує число кореляцій на завершальній стадії розпізнавання [2].

Найбільш важливою функцією нормалізації є автоматизований обрахунок невідомих параметрів перетворень, яким піддаються саме початкові образи, а потім приведення їх вже до еталонної форми. Перетворення виконується за допомогою операторів нормалізації, так званих нормалізаторів, а обрахунок параметрів здійснюється за допомогою функціоналів, що діють одночасно на багато образів [2].

1.3 Огляд існуючих методів і систем для розпізнавання підпису

Сьогодні переважна кількість OCR-програм успішно працює з різноманітними растровими картинками, що були отримані за допомогою цифрової фотокамери, сканеру, тощо [3].

Головним призначенням різних OCR-систем вважається аналіз інформації, поданої в растровому вигляді, так званий від сканований символ, а також призначення деякому фрагменту рисунка відповідного символу. Відразу після закінчення процесу розпізнавання OCR-програми повинні бути в змозі зберегти форматування результуючих документів, призначити атрибут абзацу в необхідному місці, зберегти довільні графіки, різноманітні таблиці, тощо. Новітнє програмне забезпечення може підтримувати такі формати: PDF, HTML, електронні таблиці, тощо [3].

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		16

Як правило, використання різноманітних OCR-систем, ніяким чином не викликає якихось труднощів. Переважна кількість цих програм володіє досить простим автоматизованим режимом, спрямованим на коректне сканування й розпізнавання образів, зокрема, із зовнішніх файлів [3].

Одна із найважливіших функцій такого програмного забезпечення - зручність обрання мови, що буде використовуватись під час розпізнавання, а також типу матеріалу, який слід розпізнати (факс, друкарська машинка, паперова чи електронна газета, крапковий матричний принтер, тощо). В процесі розпізнавання текстів, що використовують кілька мов одночасно, ефективність самого розпізнавання прямопропорційно залежить від здатності програми формувати групи мов. Сьогодні деякі програмні забезпечення вже мають комбінації для найбільш часто використовуваних мов, наприклад, англійської чи російської [3].

На даний момент існує неймовірна кількість різноманітного програмного забезпечення, що базується на OCR та має приблизно однаковий функціонал в результаті, тому розглядати кожне окремо немає ніякого сенсу та необхідності.

Таким чином, було прийнято рішення зобразити порівняльну характеристику найбільш відомих OCR-програм за допомогою таблиці 1.4, звертаючи увагу на особливості функціоналу кожної [3].

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		17

Таблиця 1.4 – Порівняльна характеристика OCR-програм

Назва	Ліцензія	Операційні системи	Особливості
ONLINEOCR.ru	Комерційна	Online	Online OCR сервіс дозволяє розпізнати багатомовний текст з відсканованого документа або фотографії. Конвертує результат в редаговані формати (PDF, DOC, EXCEL, TXT, HTML).
img2txt.ru	Комерційна	Online	Online OCR сервіс дозволяє розпізнати багатомовний текст з відсканованого документа або фотографії. Конвертує результат в редаговані формати (RTF, TXT, HTML).
ABBYY FineReader	Комерційна	Windows; Linux, Mac OS	Для роботи з різними мовами потрібна підтримка відповідної мови.
OmniPage	Комерційна	Windows, Mac OS	Розробники стверджують, що їх програма практично з 100% точністю розпізнає друкарські документи, відновлюючи їх форматування.
Readiris	Комерційна	Windows, Mac OS	Виробництво бельгійської I.r.i.s. Group. Містить регіональні пакети для розпізнання азіатських мов і мов середнього сходу.
Persian Reader	Комерційна	Windows	Спеціалізується на перській мові (фарсі).
Kirtas Technologies Arabic OCR	Комерційна	Windows	Може розпізнавати арабські і англійські символи на одній сторінці.

Назва	Ліцензія	Операційні системи	Особливості
CuneiForm	BSD	Windows (с GUI), Linux, Mac OS, FreeBSD (CLI)	Промислова, багатомовна система, уміє зберігати форматування тексту і розпізнає заплутані таблиці довільної структури.
Brainware	Комерційна	Windows	Імпортування даних з документів і їх обробка; наприклад рахунки, сповіщення, накладні і платежі.
HOOCR	GPL	Linux	Розпізнавання текстів на івриті.
ReadSoft	Комерційна	Windows	Сканування, розпізнавання і класифікація ділових паперів наприклад договорів, рахунків і платіжних доручень.
SILVERCODERS OCR Server	Комерційна	Linux	Серверна багатомовна система, має високу якість розпізнавання, може зберігати форматування тексту і розпізнає заплутані таблиці довільної структури.
SmartScore	Комерційна	Windows, Mac OS	Для розпізнавання нотного запису.
Tesseract	Apache	Windows, Mac OS X, Linux, OS/2	Розробляється компанією Google.

1.4 Розпізнавання тексту за допомогою нейронних мереж

В наш час штучні нейронні мережі широко використовуються для розпізнавання символів. Сьогодні більшість алгоритмів, які використовуються різними нейронними мережами, зазвичай, будуються наступним чином. Зображення символу, іншими словами растр, що є початковою картинкою для розпізнавання, зменшується до деякого стандартно визначеного розміру. Найбільш часто значення цього розміру дорівнює 16x16 пікселів [4].

У вигляді початкових параметрів нейронної мережі, зазвичай, використовуються значення яскравості у вузлах нормованого зображення символу. Кількість вихідних параметрів будь-якої нейронної мережі набуває значення рівного кількості розпізнаних символів. Найбільше значення вихідного вектору мережі відповідає конкретному символу, що, в свою чергу, і є результатом розпізнавання. Зауважимо, що ускладнення самої будови мережі чи пошук більш інформативних початкових особливостей, можуть призвести до підвищення надійності алгоритмів розпізнавання [4].

В якості вхідних параметрів нейронної мережі замість значень яскравості у вузлах нормованого зображення символу повною мірою можна використовувати значення, що характеризують різницю яскравості, іншими словами перепад. Саме завдяки таким початковим ознакам стає можливим покращення виділення межі довільної букви. Усі об'єкти, що розпізнаються, зменшуються до розміру, значення якого дорівнює 16x16 пікселів. Потім з метою вибору областей з найбільшими різницями в яскравості, дані об'єкти обробляються ще додатково.

Зауважимо, що застосування декількох елементів розпізнавання одночасно, а також подальше об'єднання одержаних результатів, може спричинити доволі значне підвищення точності розпізнавання [4].

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		20

Проте, слід пам'ятати те, щоб алгоритми, які використовуються даними елементами, були максимально незалежними. Такого ефекту можна досягти, по-перше, за допомогою самих же модулів розпізнавання, які, в свою чергу, застосовують кардинально різні алгоритми розпізнавання, по-друге, за допомогою вибору особливих навчальних даних [4].

Процес навчання нейронної мережі проходить доволі зрозуміло та просто – за допомогою алгоритму зворотного розповсюдження похибок. Спершу навчальна програма приймає спеціальний файл, в якому розміщуються растри. Далі під час навчання усі символи без виключень із файлу рухаються циклічно. Потім для кожного рисунка, без виключень, із вхідного файлу, що згадувався вище, вибираються основні особливості. Відразу ж після цього кроку здійснюється пряме та зворотне проходження по нейронній мережі. Далі після кожного символу без виключення під час навчання здійснюється зміна ваг мережі. Зауважимо, що крок модифікації ваги нейронної мережі є константним [4].

Усі символи, що погано розпізнаються переглядаються частіше за інші. Такий механізм дозволяє забезпечити вдосконалення та в результаті й прискорення всього процесу навчання. Зауважимо, що в такій ситуації слід користуватися кешом, що здатен зберігати всі складно розпізнавані образи. Слід зазначити, що рисунки для навчання можуть вибиратися або із вхідного файлу, або ж із кеша. Обрання растру із кеша ґрунтується, передусім, на якості його розпізнавання. Таким чином, отримуємо ситуацію за якої усі образи, що розпізнаються погано, будуть ще частіше обиратися [4].

Отже, під час різних стадій навчання буде цілком доречно використовувати наступні оптимізуючі чинники, що наведені нижче:

1. розмір кеша, що використовується для збереження поганих образів;
2. крок модифікації мережових коефіцієнтів;
3. в деяких ситуаціях доречно застосовувати додатковий шум;

4. вибір зображення символів із вхідного файлу, а також із кеша, що використовується для збереження поганих образів з відносною частотою;
5. історія мережевого навчання;
6. застосування мережевої регуляризації;
7. вибір коректного моменту зупинення навчання мережі [4].

Зауважимо, що будь-який алгоритм можна значно покращити, знайшовши більш влучне представлення структурних особливостей растрів, що розпізнаються. Також слід пам'ятати, що покращення якості розпізнавання можна цілком досягти за рахунок збільшення пам'яті самої мережі, або ж за рахунок застосування більшої бази інформації для навчання. Отже, розроблене програмне забезпечення повинне працювати в режимі, що є досить близьким до реального часу, і тому програмований алгоритм повинен бути доволі швидким, а з іншого боку, ще й досить коректно розпізнавати різні зображення символів [4].

Висновки до розділу

Увесь процес розпізнавання растрів складається з багатьох етапів та є водночас неймовірно складним навіть в наш час. Значна кількість етапів пов'язана, перш за все, з тим, що різні завдання обробки фактично досить тісно пов'язані між собою і якість розв'язання однієї з них впливає на вибір методу розв'язання інших. Таким чином, вибір алгоритму розпізнавання, перш за все, залежить від конкретних умов представлення початкових образів, зокрема від характеру фону, шуму і пов'язаний з вибором методів попередньої обробки, сегментації, фільтрації.

В даному розділі розглянуто тему аналізу тексту та зображень. Завдяки цьому було закріплено й одночасно поглиблено теоретичні, а також практичні знання в галузі комп'ютерних систем тестування, зокрема з розпізнаванням підпису.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		22

2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПISУ

2.1 Опис комп'ютерної системи тестування з розпізнаванням підпису

Однією з відмінних особливостей систем штучного інтелекту в звичайній, всім зрозумілій, постановці, перш за все, є застосування довільної мови, що базується на символах для представлення загальних знань про галузь предмету, а також конкретних знань про те, як вирішити проблему. Таким чином, центральним моментом у проектуванні таких систем є презентація знань, їх обробка, а також роз тлумачення [2, 3, 4].

Сьогодні технологія комп'ютерного зору активно застосовується в медицині, робототехніці, сфері безпеки та в багатьох інших областях. Завдання розпізнавання образів відноситься до області комп'ютерного зору. Нейронні мережі активно застосовують для вирішення завдань розпізнавання образів.

Компанія Google розробила нейронну мережу для розпізнавання малюнків. Для навчання мережі вони запустили онлайн-гру «Quick, Draw!», Де користувачеві ставиться завдання намалювати якийсь предмет, в процесі промальовування зображення комп'ютер намагається вгадати, що ви хочете намалювати, таким чином створюється навчальна пара: зображення і його назва. При навчанні враховується не тільки сам малюнок, а й порядок промальовування частин зображення. На базі аналогічної нейронної мережі існує Google перекладач рукописного тексту, де введення тексту може здійснюватися за допомогою промальовування тексту на екрані смартфона.

Практичним завданням даної дипломної роботи було розробити комп'ютерну систему для розпізнавання письмових символів. Центральне призначення OCR-систем в такому випадку – проаналізувати відсканований

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		23

символ, тут мається на увазі растрову інформацію, а також призначити деякому фрагменту рисунка відповідний символ. Сучасне програмне забезпечення для розпізнавання обов'язково зобов'язане підтримувати, по-перше, формати електронних таблиць, по-друге, різноманітні текстові та графічні формати [3, 4].

Одна із найважливіших функцій такого програмного забезпечення - зручність обрання мови, що буде використовуватись під час розпізнавання, а також типу матеріалу, який слід розпізнати. В процесі розпізнавання текстів, що використовують кілька мов одночасно, ефективність самого розпізнавання прямопропорційно залежить від здатності програми формувати групи мов. Сьогодні деякі програмні забезпечення вже мають комбінації для найбільш часто використовуваних мов [2, 3, 4].

У даній роботі пропонується використати багатошарову нейронну мережу з готовим навчальним набором. Такий метод дозволяє забезпечити швидкий і точний процес розпізнавання підпису користувача.

Перевагами використання саме цього методу є:

- 1) навчання на готовому наборі прикладів;
- 2) рішення завдань при невідомих закономірностях;
- 3) стійкість до шумів у вхідних даних;
- 4) надвисока швидкодія за рахунок використання масового паралелізму обробки інформації.

Сьогодні найбільш відомі архітектури нейронних мереж, а саме: карти Кохонена, нейромережі Хопфілда, а також багатошарові персептрони, - повною мірою здатні розв'язувати неймовірну кількість різноманітних задач, зокрема ті, що ніяким чином не вирішуються за допомогою звичайних методів обробки інформації [3, 4].

Розроблений метод складається із шести етапів:

- етап навчання мережі, використовуючи готовий набір;

- етап класифікації образів;
- етап кластеризації;
- етап апроксимації функцій;
- етап прогнозування;
- етап розпізнавання.

На рисунку 2.1 зображено узагальнену схему функціонування методу розпізнавання підпису за допомогою нейронної мережі.

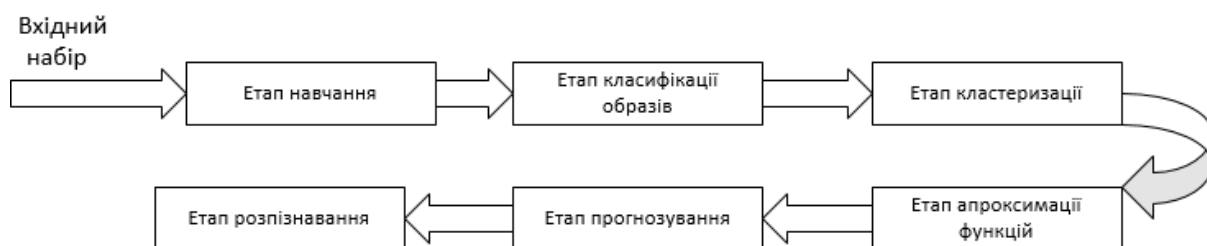


Рисунок 2.1 – Узагальнена схема функціонування нейронної мережі для розпізнавання підпису

Навчальний набір даних для нейромережі доцільніше взяти готовий, тому що у відкритому доступі є бази знань для навчання нейронних мереж. Ми використовували набір даних, що складається з чорно-білих зображень букв українського алфавіту і цифр у вигляді файлів формату PNG, написаних різними почерками. Такий набір складався за допомогою анкетування, де респондентам треба було написати всі 32 букви українського алфавіту та 10 цифр своїм звичайним почерком. Зображення букв розмірністю 64х64 пікселів перетворені в матриці за допомогою функції `imread` і за допомогою функції `reshape` з матриці був сформована матриця розміром 1х21025, що складається з нулів і одиниць. Нулі відповідають білому тлу, а одиниці - чорному. Використовувалося 6090 зображень.

Далі всі зображення були записані в одну матрицю X розмірністю 6090×21025 . Позначення класів, тобто букв, представлені в таблиці. На підставі цих позначень був сформований вектор Y розмірністю 6090×1 , який складається з чисел від 1 до 32.

Рядок n вектора Y відповідає рядку n матриці X , тобто кожній букві відповідає певне зображення. X і Y були збережені в файл розширенням `.mat`. Приклади вхідних символів зображено на рисунку 2.2.



Рисунок 2.2 – Приклади вхідних символів

Для розпізнавання зображень рукописних букв використовується тришарова нейронна мережа.

Вхідний шар нейронної мережі має 21025 нейронів, так як в якості навчального набору використовуються зображення рукописних букв дозволом 64×64 пікселів, тобто зображення складається з 21025 пікселів. Таким чином вхідним сигналом для нейронної мережі є кожен піксель зображення.

Вихідний шар має 42 нейронів, тому що використовується 32 букви українського алфавіту і 10 цифр. Для прихованого шару важливо вибрати оптимальну кількість нейронів. Якщо вибрати дуже маленьку кількість

нейронів, то мережа може бути недовченою, при виборі занадто великої кількості нейронів - перенавченою. У таких випадках нейромережа не зможе правильно передбачати значення. Можна скористатися правилом геометричної піраміди, тоді слід вибрати 780 нейронів. Відповідність букв та їх значення наведені в таблиці 2.1.

Таблиця 2.1 – Відповідність букв і їх значення

Клас (Буква)	Значення класу	Клас (Буква)	Значення класу
А	1	О	18
Б	2	П	19
В	3	Р	20
Г	4	С	21
Д	5	Т	22
Е	6	У	23
Є	7	Ф	24
Ж	8	Х	25
З	9	Ц	26
И	10	Ч	27
І	11	Ш	28
Ї	12	Щ	29
Й	13	Ь	30
К	14	Ю	31
Л	15	Я	32
М	16		
Н	17		

Центральний елемент розроблюваної комп'ютерної системи – багатошаровий (видозмінений) персептрон, зображений на рисунку 2.3.

Розширення розпізнаваних зображень – це число синапсів (N). Число растрів (M), що навчаються за допомогою персептрона – це кількість нейронів цього ж персептрона [5].

Як приклад, проаналізуємо алгоритм зворотного розповсюдження помилки для обраного персептрона. Усі символи, без виключення, зображуються у вигляді картинок, значення розмірів яких, наприклад, дорівнюють 79,38x52,92 мм. В нашому випадку всі образи мають нестандартне розширення .bmp та бітовий тип. Кожен растр поділяється на 600 рівних частин за допомогою нейронної мережі (30 по вер. та 20 по гор.). Таким чином, число синапсів є константним для кожного нейрона та обраховується за наступною формулою [5]:

$$n = 20 * 30 = 600$$

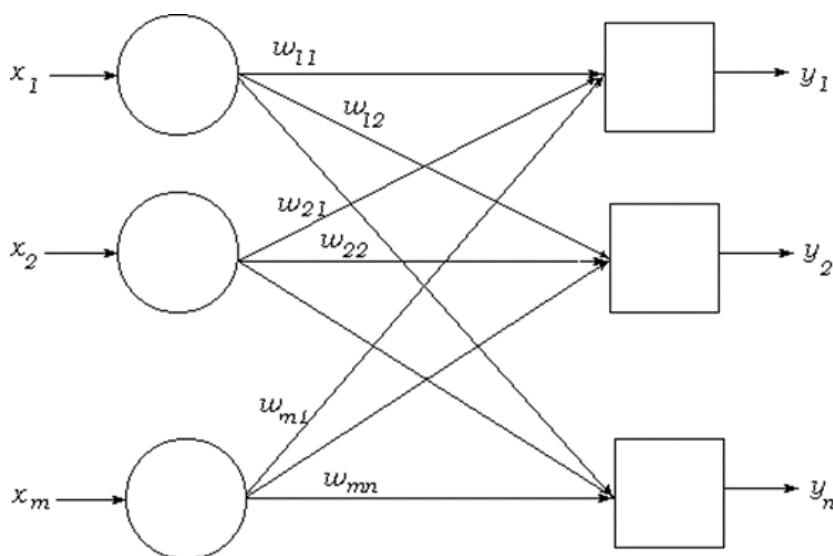


Рисунок 2.3 – Узагальнена структура видозміненого персептрона

Навчання обраного персептрона з урахуванням запропонованого підходу можна подати наступним чином [5]:

1. усім вагам нейронної мережі, без виключення, присвоюються неймовірно малі, майже рівні нулю, значення;

2. початковий вектор X для навчання, що складається із вхідних елементів подається безпосередньо на вхід нашої мережі;
3. значення величини, тут мається на увазі сигнал, NET результуючої функції від кожного нейрона обраховується за наступною формулою [5]:

$$NET_j = \sum_i x_i w_{ij}$$

де x_i - i -ий елемент вхідного вектора X ; w_{ij} - вага входу i нейрона j ; NET_j – величина збудження нейрона після взаємодії з синапсом;

4. величина граничної функції стимуляції сигналу NET від кожного нейрона обраховується за наступними формулами [5]:

$$OUT_j = 1, \text{ якщо } NET_j \text{ більш ніж поріг } u_j$$

$$OUT_j = 0, \text{ в іншому випадку}$$

де u_j являє собою поріг, що відповідає нейрону j , наприклад, для тривіального випадку усі нейрони, без виключення, матимуть однаковий поріг;

5. похибка кожного нейрона обраховується за наступною формулою [6]:

$$error_j = target_j - OUT_j$$

де $error_j$ - помилка для j -го нейрона; $target_j$ - необхідний вихід j -го нейрона; OUT_j - отриманий вихід j -го нейрона;

6. кожна вага модифікується за наступною формулою [5, 6]:

$$w_{ij}(t + 1) = w_{ij}(t) - \delta x_i error_j$$

де $w_{ij}(t + 1) = w_{ij}$ - вага входу i нейрона j в момент часу $(t + 1)$; $w_{ij}(t)$ - вага входу i нейрона j в момент часу t ; δ – це нормуючий параметр. Його вхідне значення складає 0,1. Протягом навчання значення даного параметра зменшується [6].

Поки похибка достатньо не зменшиться, етапи з третього по шостий будуть повторятися. Мета центрального модуля комп'ютерної системи тестування з розпізнаванням підпису – підрахунок виходів персептрона [6].

Функція стимуляції кожного нейрона, без виключення, наведена нижче:

$$Y = S, \text{ при } S < T,$$

$$Y = S, \text{ при } S \geq T,$$

де Y – величина функції; S – її аргумент; T - значення порога.

Далі під час функціонування інтелектуального модуля, що є центральним в комп'ютерних системах тестування з розпізнаванням підпису відбувається подальше навчання персептрона [6].

Модифікація ваг усіх синапсів, без виключення, відбувається за формулою:

$$\delta = (D - Y)$$

де δ – це значення, що дорівнює різниці між потрібним виходом D і реальним виходом Y .

Отже, розпізнавання образів буде відбуватися за наступним алгоритмом [6]:

1. при запуску програмного забезпечення відбувається завантаження налаштувань за замовчуванням;
2. утворення растрів, що є еталонними;
3. утворення або ж завантаження растрів, що будуть розпізнаватися;
4. процес розпізнавання образу, що має зашумленість менше 40%;
5. формування вихідного результату.

Для наглядності запропонований алгоритм розпізнавання зображено на рисунку 2.4.

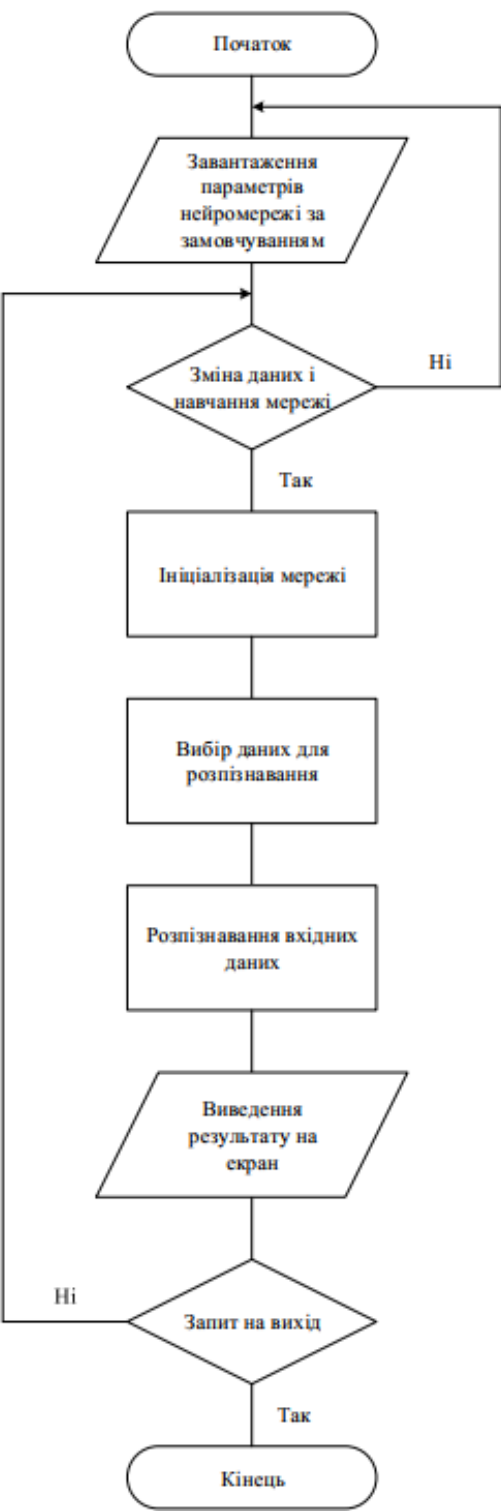


Рисунок 2.4 – Схема алгоритму нейронної мережі для розпізнавання підпису

2.2 Проектування комп'ютерної системи з розпізнаванням підпису за допомогою нейронної мережі

Для проектування всіх етапів запропонованого методу було проведено експеримент за допомогою бібліотеки OpenCV [7].

OpenCV (Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим вихідним кодом) - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних і комерційних цілях - поширюється в умовах ліцензії BSD [7].

Для експерименту використовуємо навчальну вибірку даних наведену на рисунку 2.5.

9821480865132823066470938
4460955058223172535940812
8481117450284102701938521
1055596446229489549303819
6442881097566593344612847

Рисунок 2.5 – Вибірка даних для навчання нейромережі

Далі відбувається підготовка даних до навчання за допомогою наступної програми, що зображена на рисунку 2.6.

```
import sys
import numpy as np
import cv2
im = cv2.imread('pitrain.png')
im3 = im.copy()
gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray,(5,5),0)
thresh = cv2.adaptiveThreshold(blur,255,1,1,11,2)
##### Now finding Contours #####
contours,hierarchy = cv2.findContours(thresh,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
samples = np.empty((0,100))
responses = []
keys = [i for i in range(48,58)]
for cnt in contours:
    if cv2.contourArea(cnt)>50:
        [x,y,w,h] = cv2.boundingRect(cnt)
        if h>28:
            cv2.rectangle(im,(x,y),(x+w,y+h),(0,0,255),2)
            roi = thresh[y:y+h,x:x+w]
            roismall = cv2.resize(roi,(10,10))
            cv2.imshow('norm',im)
            key = cv2.waitKey(0)
            if key == 27: #(escape to quit)
                sys.exit()
            elif key in keys:
                responses.append(int(chr(key)))
                sample = roismall.reshape((1,100))
                samples = np.append(samples,sample,0)
responses = np.array(responses,np.float32)
responses = responses.reshape((responses.size,1))
print "training complete"
np.savetxt('generalsamples.data',samples)
np.savetxt('generalresponses.data',responses)
```

Рисунок 2.6 – Підготовка даних до навчання

Дана програма виконує такі функції [8]:

- завантажує зображення;
- вибирає цифри (очевидно, шляхом пошуку контурів і застосування обмежень по площі і висоті букв щоб уникнути помилкових спрацьовувань);

- малює обмежує прямокутник навколо однієї літери;
- після натискання відповідних клавіш він змінить цей розмір на 16×16 і збереже 100 значень пікселів в масиві і відповідну введену вручну цифру в іншому масиві.
- потім збережіть обидва масиви в окремих файлах txt [8].

Результат виконання першого етапу програми зображено на рисунку 2.7

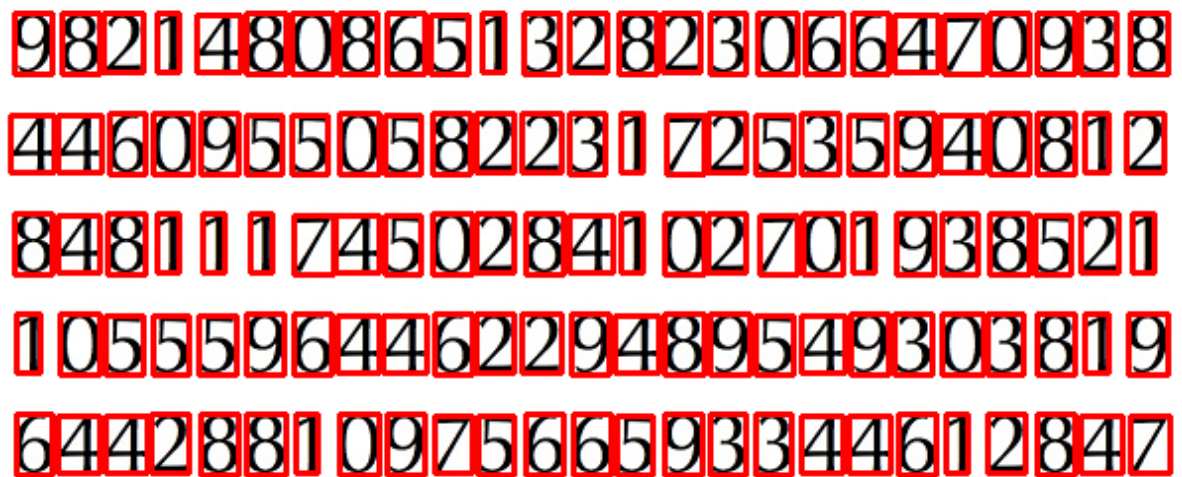


Рисунок 2.7 – Результат виконання першого етапу програми

Для тестування роботи програми розпізнавання візьмемо іншу вибірку і проводимо навчання мережі. Далі ми робимо наступні дії [8]:

- завантажуюємо файл txt, який ми зберегли раніше
- створюємо екземпляр класифікатора, який ми використовуємо (KNearest)
- використовуємо функцію KNearest.train для навчання даних

Вибірку даних для тестування процесу навчання мережі зображено на рисунку 2.8.

3.141592653589793238462643
 3832795028841971693993751
 0582097494459230781640628
 6208998628034825342117067
 9821480865132823066470938
 4460955058223172535940812
 8481117450284102701938521
 1055596446229489549303819
 6442881097566593344612847

Рисунок 2.8 – Вибірка даних для тестування процесу навчання мережі

В процесі тестування ми робимо наступні дії [8]:

- завантажуюмо зображення, яке використовується для тестування
- обробляємо зображення як раніше і витягуємо кожну цифру, використовуючи контурні методи
- намалюйте обмежувальну рамку для нього, потім змініть розмір до 16×16 і збережіть його значення пікселів в масиві, як це було зроблено раніше.
- потім ми використовуємо функцію `KNearest.find_nearest()`, щоб знайти найближчий елемент до того, який ми подали [8].

Процес тестування навчання мережі за допомогою програми зображено на рисунку 2.9.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		35


```

import cv2
import numpy as np
##### training part #####
samples = np.loadtxt('generalsamples.data',np.float32)
responses = np.loadtxt('generalresponses.data',np.float32)
responses = responses.reshape((responses.size,1))
model = cv2.KNearest()
model.train(samples,responses)
##### testing part #####
im = cv2.imread('pi.png')
out = np.zeros(im.shape,np.uint8)
gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
thresh = cv2.adaptiveThreshold(gray,255,1,1,11,2)
contours,hierarchy = cv2.findContours(thresh,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    if cv2.contourArea(cnt)>50:
        [x,y,w,h] = cv2.boundingRect(cnt)
        if h>28:
            cv2.rectangle(im,(x,y),(x+w,y+h),(0,255,0),2)
            roi = thresh[y:y+h,x:x+w]
            roismall = cv2.resize(roi,(10,10))
            roismall = roismall.reshape((1,100))
            roismall = np.float32(roismall)
            retval, results, neigh_resp, dists = model.find_nearest(roismall, k = 1)
            string = str(int((results[0][0])))
            cv2.putText(out,string,(x,y+h),0,1,(0,255,0))
cv2.imshow('im',im)
cv2.imshow('out',out)
cv2.waitKey(0)

```

Рисунок 2.9 – Процес тестування навчання мережі

В результаті виконання програми отримаємо графічне зображення на рисунку 2.10. Проектування системи розпізнавання підпису було згідно розробленого алгоритму методу розпізнавання підпису за допомогою багат шарової нейронної мережі.

3	14	15	92	65	35	89	79	32	38	46	26	43
38	32	79	50	28	84	19	71	69	39	93	75	1
05	82	09	74	94	45	92	30	78	16	40	62	8
62	08	99	86	28	03	48	25	34	21	17	06	7
98	21	48	08	65	13	28	23	06	64	70	93	8
44	60	95	50	58	22	31	72	53	59	40	81	2
84	81	11	74	50	28	41	02	70	19	38	52	1
10	55	59	64	46	22	94	89	54	93	03	81	9
64	42	88	10	97	56	65	93	33	44	61	28	4
7												

Рисунок 2.10 - Відображення результатів тестування нейромережі

Висновки до розділу

Запропоновано метод розпізнавання підпису на основі багатошарових нейронних мереж . Метод дозволяє забезпечити введення та обробку вхідних даних користувача, також логічне виведення інформації.

Запропонований метод покращить розпізнавання введеної інформації та допоможе удосконалити процес навчання мережі, згідно вже відомих параметрів.

3 РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ З РОЗПІЗНАВАННЯМ ПІДПISУ

3.1 Вибір типу архітектури програмного забезпечення та методу проектування

Вибір найбільш вигідної і «хорошої» архітектури для свого ПЗ є важливим і потрібним кроком. Для вибору архітектури ми слідували списку таких критеріїв:

1. Ефективність системи - насамперед програма, безумовно, повинна вирішити поставлену проблему і реалізовувати свої функції відмінно, і за різних обставин. Найбільш важливі характеристики: продуктивність, надійність, здатність справлятися зі збільшенням навантаження, безпека, тощо.

2. Гнучкість системи - будь-яка програма повинна змінюватися з часом в результаті зміни вимог або ж при появі нових. Чим оперативніше та комфортніше користувач зможе внести правки в поточний функціонал, тим менше різних похибок та негараздів це спричинить. Як результат, на виході отримаємо конкурентоздатне програмне забезпечення. Тому під час проектування будь-якої програми слід оцінити, що отримаємо, з точки зору того, як це може потім з легкістю змінитися. Модифікація однієї частини програмного забезпечення ніяким чином не повинна мати вплив на всі інші його частини. Там, де це можливо, архітектурні пропозиції не повинні бути шкідливими для всієї програми загалом, а наслідки від різноманітних похибок мають бути мінімальними.

3. Можливість розширення системи - можливість додавати нові функції в систему, при цьому ніяким чином не порушуючи її початкову організацію. На одному з перших етапів розробки програмне забезпечення слід забезпечити тільки найнеобхіднішим функціоналом, але, з іншого боку, конструкція повинна дозволяти користувачу без зайвих проблем будувати додатковий функціонал за потребою. Внесення найімовірніших правок має вимагати найменших зусиль.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		38

4. Принцип відкритості / закритості - це означає, що система повинна бути спроектована таким чином, щоб зміна її поведінки, а також збільшення кількості її функцій було досягнуто саме шляхом написання нового програмного коду, не змінюючи початковий код. Як результат, при надходженні нових вимог, наприклад від заказчика – початкова логіка програми ніяким чином не зміниться.

5. Масштабування процесу розробки - це здатність зменшити час розробки шляхом долучення до проєкту нових програмістів. Архітектура повинна забезпечувати паралельний процес розробки, щоб декілька програмістів могло працювати над розроблюваною системою одночасно.

6. Повторність використання – бажано розроблювати програмне забезпечення таким чином, щоб його складові могли бути повторно використані в інших програмах.

7. Технічне обслуговування – система має бути відмінно структурованою, не дублюватися, володіти гарним програмним кодом, а також документацією. Зауважимо, що під час розробки програмного забезпечення слід застосовувати якісь загальноприйняті рішення, відомі та знайомі для більшості програмістів.

Незважаючи на різноманітність критеріїв, головне завдання в розробці програмного забезпечення - зменшити складність. А щоб зменшити складність, нічого кращого та більш ефективного, крім розбиття програми на довільну кількість фрагментів, ще не винайдено.

На просторах Інтернету такий підхід часто називають, як принцип "розділяй та володарюй" (divide et impera), але насправді це звичайна ієрархічна декомпозиція. Громіздку програму слід будувати за допомогою невеликої кількості менш громіздких підпрограм, кожна з яких, у свою чергу, будується ще з менших фрагментів, тощо, до того моменту, поки найдрібніші фрагменти не будуть простими, щоб безпосередньо зрозуміти і створити їх власноруч.

Для розроблюваного ПЗ передбачено використання Use-case діаграми,

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		39

показаної на рисунку 3.1. Зауважимо, що Use-case діаграми описують лише відношення і залежності між різними варіантами використання та користувачами, які безпосередньо беруть участь у відповідному процесі.



Рисунок 3.1 – Загальна діаграма використання ПЗ

Слід зазначити, що Use-case діаграми не доцільно використовувати для представлення будь-якого проєкту, бо вони не здатні описати внутрішню структуру ПЗ. Такі діаграми розроблені для полегшення взаємодії з майбутніми користувачами ПЗ, із замовниками, і особливо будуть потрібні для визначення потрібних особливостей ПЗ. Таким чином, Use-case діаграми демонструють, що ПЗ повинне виконувати, не вказуючи використовувані методи.

Зауважимо, що Use-case діаграма - це опис типових взаємодій між користувачем ПЗ та самим ПЗ безпосередньо. Такі діаграми, перш за все, демонструють зовнішній інтерфейс ПЗ, а також вказують форму того, що ПЗ повинне виконувати (саме що, а не як).

Працюючи з Use-case діаграмами, бажано дотримуватись наступних правил:

1. будь-який Use-case відноситься, як мінімум, до одного користувача;
2. будь-який Use-case завжди має свого ініціатора;
3. будь-який Use-case спричиняє відповідний результат;
4. будь-який Use-case може взаємодіяти з іншим Use-case.

3.2 Розробка архітектури комп'ютерної системи тестування з розпізнаванням підпису

Основу запропонованої архітектури розроблюваного ПЗ архітектури з розпізнаванням підпису модульну схему організації взаємодії між компонентами з розгалуженою системою інших допоміжних модулів.

Для створення модуля навчання використовується класифікаційна модель, базується на встановленій навчальній вибірці і має можливість навчатися в процесі роботи, що дозволяє ефективніше виконувати функції обробки інформації та розпізнавання даних.

Архітектура системи представлена на рисунку 3.2.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		41

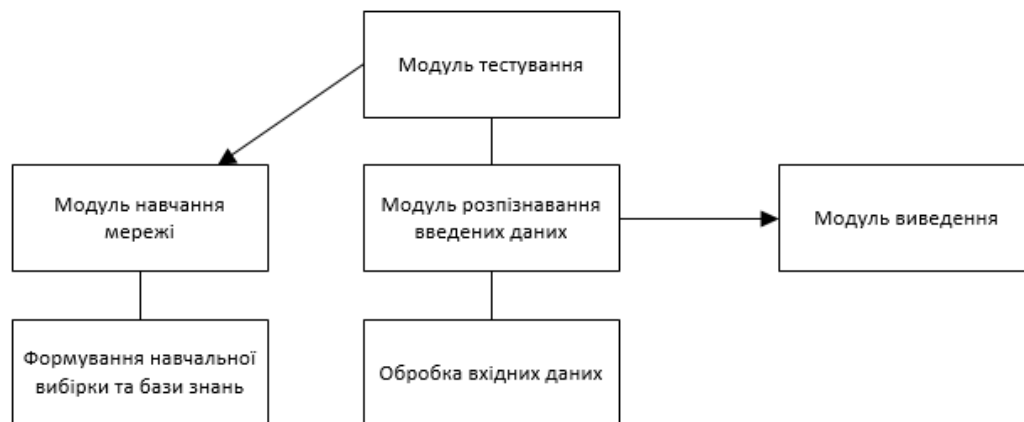


Рисунок 3.2 – Архітектура комп’ютерної системи розпізнавання підпису

Проектована комп’ютерна система складається з наступних компонентів:

- модуль тестування - призначений для тестування користувачів.
- модуль навчання – в ньому відбувається процес навчання нейронної мережі, за допомогою навчальної вибірки та бази знань;
- модуль розпізнавання введених даних – на основі введених даних сформується друкована відповідь на питання.
- модуль виведення генерує множину вихідних значень відповідей користувача;

Основою для розпізнавання підпису користувача є база знань українського алфавіту та цифр.

3.2.1 Діаграма варіантів використання для комп’ютерної системи

Основні функції використання проектного ПЗ:

- генерація бази правил для формування параметрів;
- головний актор: користувач;

- рівень взаємодії: користувач – програмний засіб;
- передумова: користувач відкрив програмний засіб.

Головні дії користувача.

1. Користувач відкриває вікно «Тестування».
2. Генерується ряд випадкових запитань.
3. Дає відповідь на ряд запитань у спеціальному вікні.
4. Всі відповіді зберігаються в окремому масиві.
5. Далі всі відповіді порівнюються з даними навчальної вибірки.
6. Розпізнавання вхідних параметрів опитування.
7. Отримуємо кількість правильних введених відповідей на тести.

Діаграма варіантів використання представлена на рисунку 3.3.

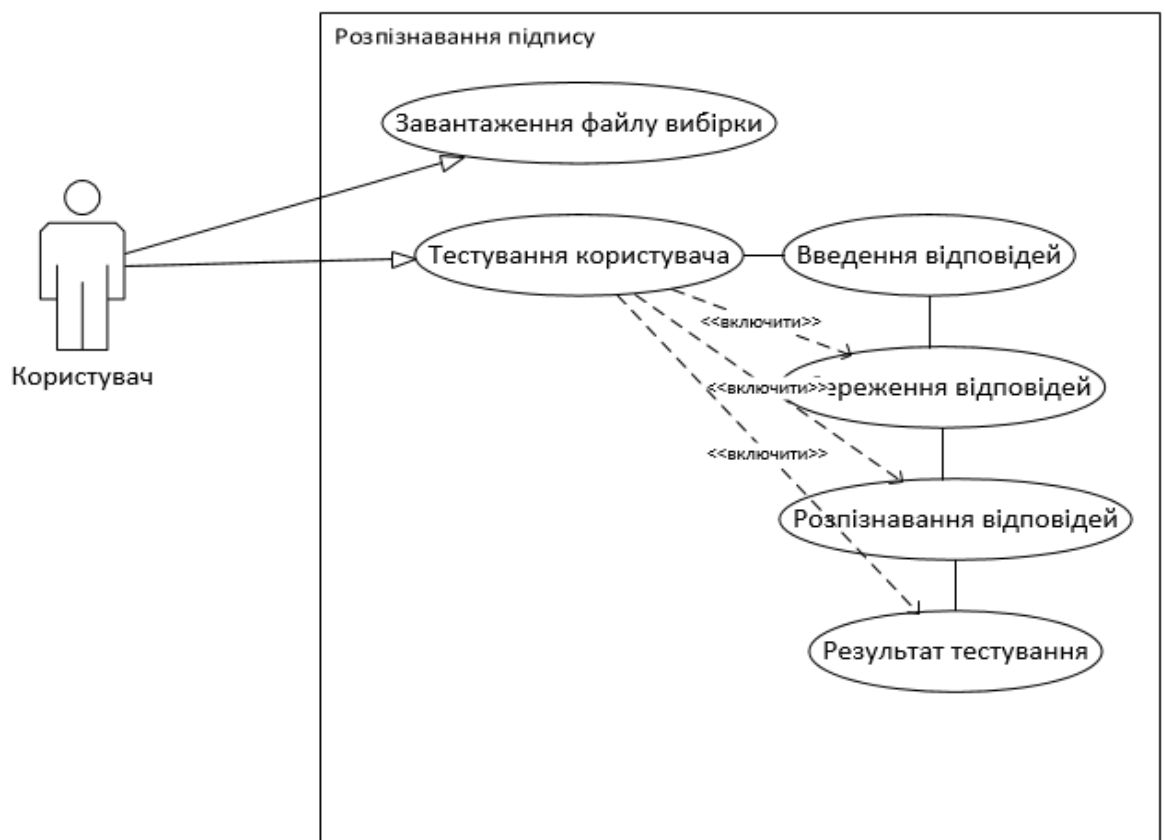


Рисунок 3.3– Uml-діаграма варіантів використання ПЗ розпізнавання підпису

3.2.2 Діаграма класів комп'ютерної системи

Програмне забезпечення складається з 3 основних модулів, кожен з яких відповідає своїм задачам:

- клас тестування;
- модуль розпізнавання відповідей;
- модуль інтерфейсу.

Діаграму класів використання ПЗ розпізнавання підпису зображено на рисунку 3.4.

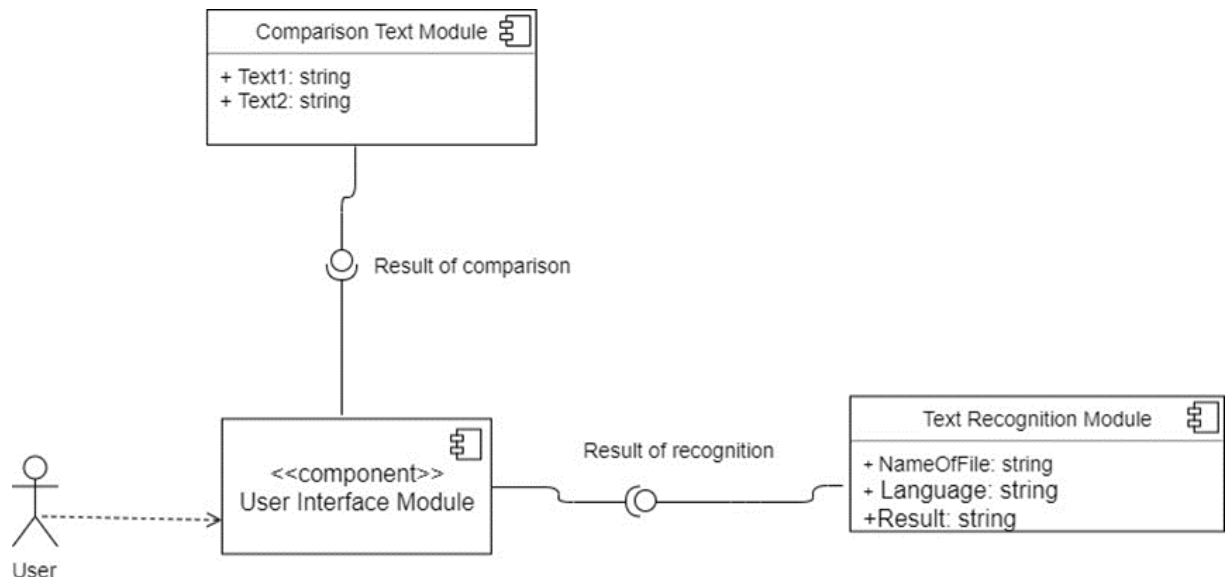


Рисунок 3.4 - Uml-діаграма класів використання ПЗ розпізнавання підпису

Під час розробки комп'ютерної системи тестування з розпізнаванням підпису було використано мову програмування C# - для створення функціональних модулів, для реалізації інтерфейсу користувача використовується Windows Form в середовищі програмування Visual Studio, а для розпізнавання - бібліотеку OpenCV [7].

В таблиці 3.1 описано всі модулі та класи, що до них входять відповідно.

Таблиця 3.1 – Функціонал програмного забезпечення

Назва модуля	Клас	Опис та призначення класу
Text Recognition Module	FileManager	Клас, що відповідає за виклик відповідного класу для зчитування інформації. Отримує назву файлу, визначає тип файлу та передає дані наступному класу.
	ReadImg	Виконує функцію розпізнавання введених символів та видає розпізнаний текст.
	ReadTextClass	Виконує функцію зчитування тексту з файлів формату txt. Отримує назву файлу і видає зчитаний текст.
	ReadWordFile	Виконує функцію зчитування тексту з файлів формату doc, txt.
User Interface Module	Form1	Відображає форму та все елементи керування, які доступні для використання. Відповідає за роботу кожного елемента та реагує на подіє, які виникають з кожним з них. Містить в собі програмну логіку.
	Program	Запускає програму, що описана в класі Form1.
Comparison Module	ComparisonTwoText	Реалізовує алгоритм навчання нейронної мережі. В результаті повертає введені дані.
	SaveResult	Зберігає введені дані користувача.

3.2.3 Діаграма станів комп'ютерної системи

Наведемо діаграму станів ПЗ і користувача (рисунок 3.5). Спочатку вказуємо дані для навчання системи та формування бази знань. Користувач проходить опитування де вказує комп'ютерною мишею свої відповіді та прізвище в спеціальному вікні. Система обробляє вхідні значення та формує відповідь в друкованому вигляді. Формується підмножина необхідних параметрів для виведення та формується кінцевий результат у вигляді кількості правильних відповідей та прізвища користувача.

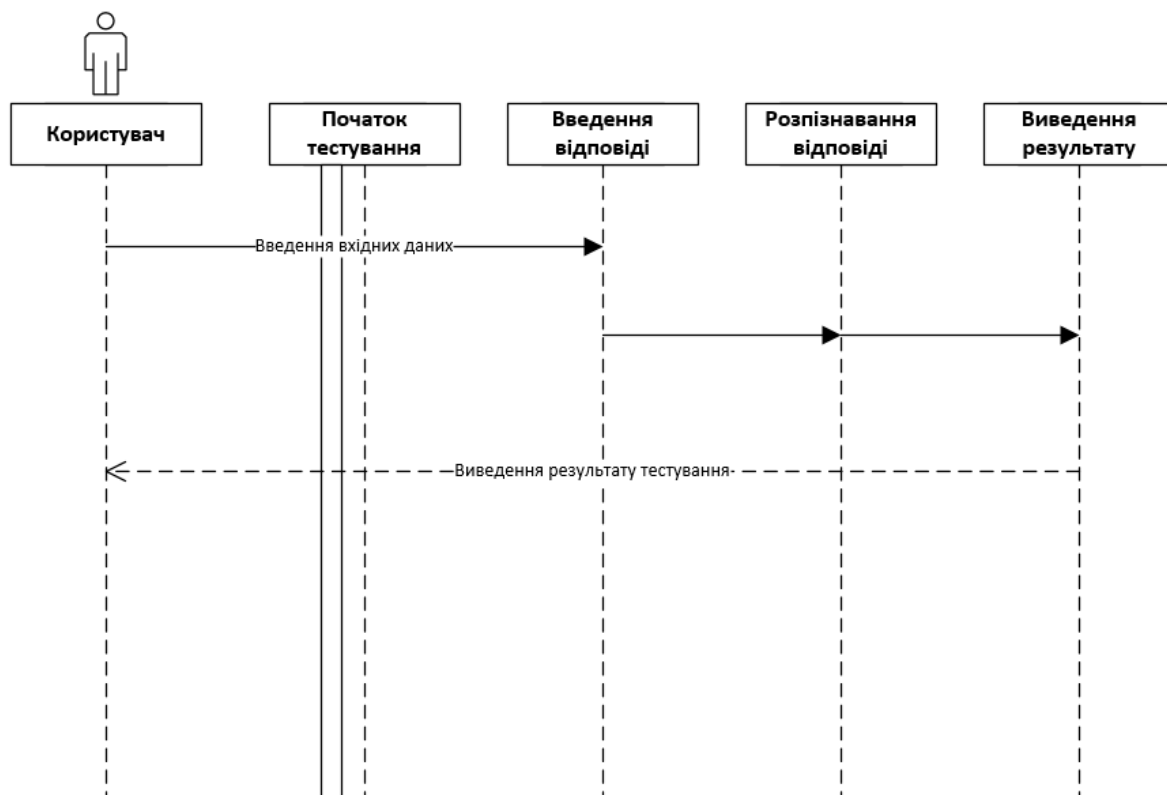


Рисунок 3.5 - Uml-діаграма послідовності використання ПЗ розпізнавання підпису

3.3 Інтерфейс комп'ютерної системи тестування з розпізнаванням підпису

Програма запускається викликом файлу «Testing.exe», після чого перед користувачем з'являється головна форма (рисунок 3.6). Якщо відразу після запуску з'явиться форма ПЗ, то програму встановлено коректно і вона успішно працює. На головній формі користувачу пропонується натиснути кнопку та почати тестування.

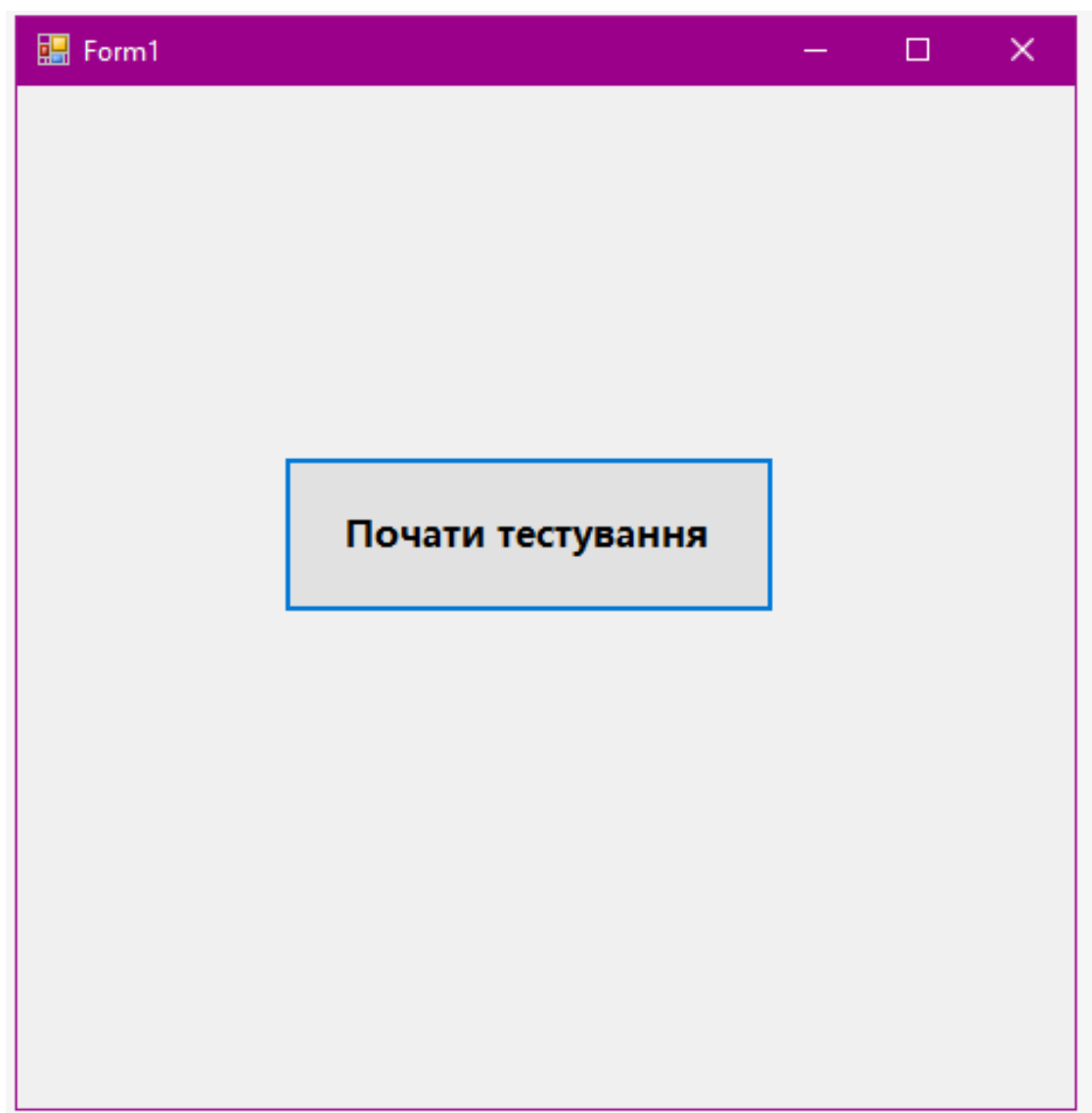


Рисунок 3.6 - Головне вікно програми

Після натиснення кнопки він отримає згенеровану випадковим чином групу тестових питань на які потрібно дати відповіді (рисунок 3.7).

Для навчальних даних використовується набір даних – база даних питань і відповідей. Користувач починає тестування та вводить відповідь на питання за допомогою комп'ютерної миші у спеціальному вікні PictureBox, після чого його відповідь розпізнається і їй присвоюється правдиве чи хибне значення.

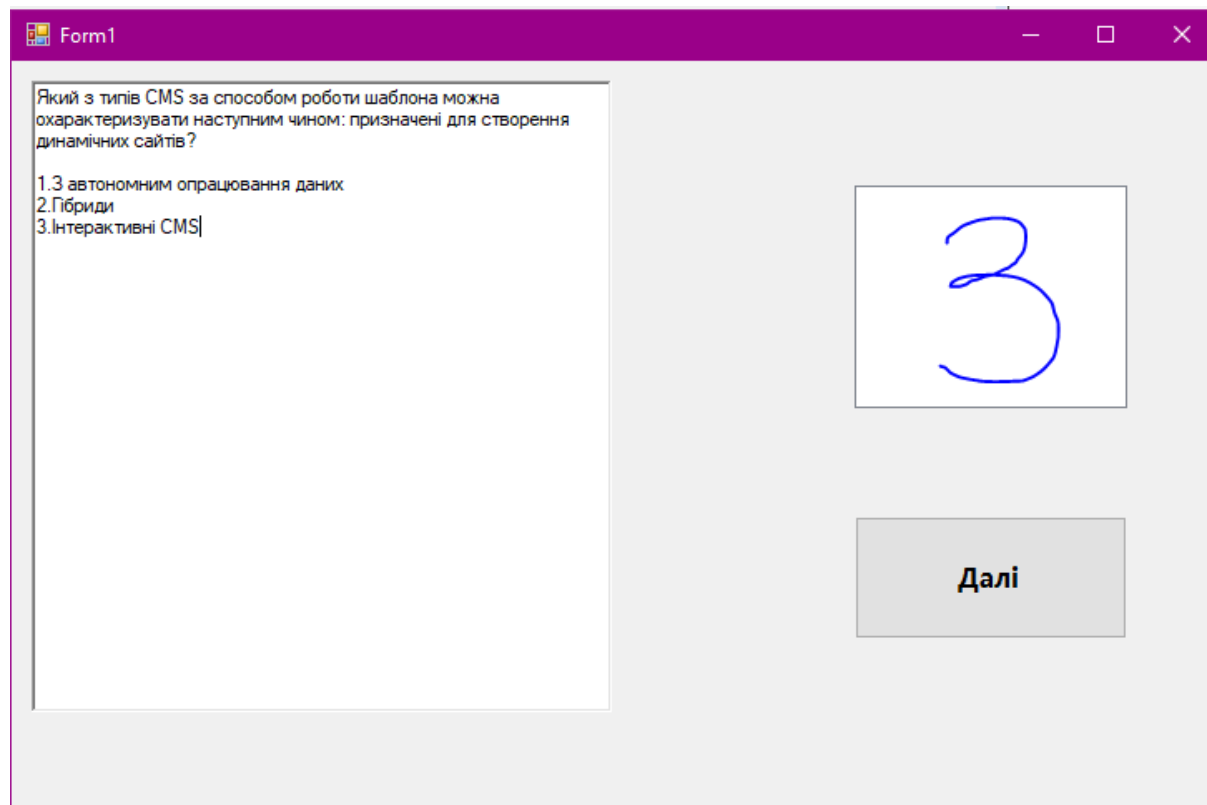


Рисунок 3.7 - Вікно процесу тестування

Після введення відповіді і натиснення кнопки «Далі», програма застосує багатoshарову нейромережу для розпізнавання символу. І коли тестові питання закінчаться потрібно буде ввести прізвище студента, щоб завершити тестування (рисунок 3.8).

Рисунок 3.8 - Вікно завершення тестування

В результаті роботи програми користувач отримує кількість правильних відповідей на запитання і власне прізвище в друкованому вигляді (рисунок 3.9).

Form1

Студент: Іванов

Кількість правильних відповідей: 3

Рисунок 3.9 - Результат роботи програми

Опис основних функцій програмного забезпечення подано в зведеній таблиці 3.2.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		50

Таблиця 3.2 – Опис основних функцій ПЗ

№ п.п	Назва функції	Опис функції
1	MainWindow(Form1 *parent)	Функція створює основне вікно ПЗ – форму, є базовим об'єктом, в якому створюються решта об'єктів
2	Setthis(this)	Функція здійснює налаштування створеного вікна
3	on_pushButton1 ()	Функція здійснює занесення в список вхідних параметрів
4	PictureBox()	Функція створює елемент-поле для введення даних
5	addSet (str)	Функція додає елемент (пункт меню, рядок, змінну)
6	elem(i)	Елемент (пункт меню, рядок, змінна)
7	on_pushButton clicked()	Функція здійснює занесення в список введених параметрів
8	OutputSetting ()	Функція здійснює виведення всіх елементів графічного інтерфейсу
9	tableWidget()	Функція формує таблицю правил
10	about(this)	Функція видає повідомлення про результат операції
11	qt_static_metacall()	Функція обробки статичних об'єктів графічного інтерфейсу
12	staticMetaObjectExtr aData()	Функція створення та оперування об'єктами графічного інтерфейсу
13	getStaticMetaObject()	Функція виклику статичних об'єктів графічного інтерфейсу

№ п.п	Назва функції	Опис функції
14	qt_metacast()	Функція передачі даних між об'єктами графічного інтерфейсу
15	qt_meta_stringdata_MainWindow()	Функція передачі та обробки даних головного вікна графічного інтерфейсу

Висновки до розділу

Увесь процес розпізнавання растрів складається з багатьох етапів та є водночас неймовірно складним навіть в наш час. Значна кількість етапів пов'язана, перш за все, з тим, що різні завдання обробки фактично досить тісно пов'язані між собою і якість розв'язання однієї з них впливає на вибір методу розв'язання інших. Таким чином, вибір алгоритму розпізнавання, перш за все, залежить від конкретних умов представлення початкових образів, зокрема від характеру фону, шуму і пов'язаний з вибором методів попередньої обробки, сегментації, фільтрації.

В даному дипломному проєкті була розглянута тематика розпізнавання підпису. В результаті виконання дипломної роботи мною була створена комп'ютерна система розпізнавання підпису за допомогою багатосарової нейронної мережі. Крім того набув навичок роботи з професійним програмним забезпеченням даного виду, закріпив і поглибив теоретичні і практичні знання в галузі програмування.

ВИСНОВКИ

В даному дипломному проекті за результатами виконаних теоретичних та практичних досліджень розроблено комп'ютерну систему тестування з розпізнавання підпису.

У першому розділі «Огляд існуючих підходів, щодо комп'ютерної системи тестування з розпізнаванням підпису» була досліджена предметна область, досліджено відомі методи, виявлені переваги і недоліки сучасних OCR-систем та відбулась постановка задачі.

У другому розділі «Проектування комп'ютерної системи тестування з розпізнаванням підпису» розроблено метод , за допомогою якого буде відбуватись розпізнавання підпису.

У третьому розділі «Реалізація комп'ютерної системи тестування з розпізнаванням підпису» відбулася розробка програмного забезпечення. На цьому етапі була вибрана архітектура; була описана декомпозиція модулів, процесів і даних.

Робота системи розпізнавання підпису здійснюється на основі багатосарової нейронної мережі реалізованої за допомогою бібліотеки OpenCV.

Нейронна мережа для обробки вхідних даних користувача, дозволяє ефективно розпізнавати вхідні ресурси для їх обробки.

В результаті виконаного дослідження набув подальшого розвитку метод розпізнавання підпису за допомогою багатосарової нейронної мережі, розроблена архітектура і компоненти програмного забезпечення, що реалізує систему розпізнавання підпису

Розроблено програмне забезпечення для створення комп'ютерної системи розпізнавання підпису.

					ІАЛЦ.045490.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		53

ПЕРЕЛІК ПОСИЛАНЬ

1. Комп'ютерні системи тестування знань та їх аналіз. *Allbest* : URL: https://revolution.allbest.ru/pedagogics/00825755_0.html (дата звернення: 23.04.2020).
2. Комплекс розпізнавання письмового тексту. *Yamiki* : URL: <https://yamiki.ru/item/25182> (дата звернення: 25.04.2020).
3. Система розпізнавання рукописного тексту. *Библиофонд* : URL: <https://www.bibliofond.ru/view.aspx?id=722342> (дата звернення: 27.04.2020).
4. Алгоритми реалізації ПЗ розпізнавання письмового тексту. *Бобродобро* : URL: <https://prog.bobrodobro.ru/29257> (дата звернення: 27.04.2020).
5. Використання нейронних мереж для розпізнавання символів. *Pandia* : URL: <https://pandia.ru/text/79/540/56880.php> (дата звернення: 03.05.2020).
6. Розпізнавання рукописних символів та зображень за допомогою нейронних мереж. *Lib* : URL: <https://ir.lib.vntu.edu.ua/bitstream/handle> (дата звернення: 05.05.2020).
7. OpenCV. *Вікіпедія* : URL: <https://uk.wikipedia.org/wiki/OpenCV> (дата звернення: 06.05.2020).
8. Використання бібліотеки OpenCV. *Stack* : URL: <https://cloud.tencent.com/developer/ask/43827> (дата звернення: 07.05.2020).